LINUX in HOUSE

THE FUN PARTS

how it's made

Introduction
Tips

MicroCenter.com
builder's dream

Not just pieces and parts

It is possible to build a computer from parts, and just as satisfying to have someone else build it. Building a PC is like working with a house builder or selecting a new car from the factory. Discovering options available and only pay for what you need will make ownership of the hardware and software more satisfying.

For example, I like small form factor systems that consume less power. Partly to save electrical costs and enable longer run-time on a battery backup (UPS) system. That's why I chose the Intel NUC [1] and BeagleBone [2] Small Board Computer (SBC) systems for light duty jobs like e-mail and cloud, and larger ATX [3] motherboards in a server case for Network Attached Storage (NAS) [4], housing up to five SATA disk drives.

1. https://www.asus.com/us/displays-desktops/nucs/all-series/
2. https://beagleboard.org
3. https://en.wikipedia.org/wiki/ATX
4. https://www.truenas.com

How to decide

The first thing I think about is what do I want to do. What computing feature of my life would I like to learn about and have control over.

Next consider having a backup plan. Like when bad weather hits, where would you go? Same thing with building your computer system(s). I generally have two pieces of hardware for each task, then document and backup the software. Backup systems can be hand-me-downs or replaced systems and parts to save money.

Finally I choose how much time and money I want to invest. Usually time equals money. Some things you may not be comfortable doing just now, and change your mind later when you have more

experience.

Why Open Source

After investing my time and money it is a shame to find out that company I bought something from has discontinued my product or gone out of business. This can also occur in Open Source environments if the developers retire, move on or just get tired of working on it.

With Open Source you can obtain the source code and keep it going yourself or find someone who can. Using your skill and knowledge while building your system will provide confidence and know how. Plus there are many resources on the Internet to ask questions, and answers along the way.

Running your services on your own system

You have the source code, you can make the change.

can provide better security as hackers like to go for bigger targets. Your data is not on someone else's disk (cloud), but your own that only you provide access to. System administrators have access to everything on the computer so be your own Sys Admin over your own data.

Advanced experience affords you the ability to change what you do not like or fix a problem the original owner does not. You have the source code, you can make the change.

Hmmmmm...

Evaluating a Project

Open Source projects will post lots of information about themselves that you can use to decide if it is worth your time to use it.

First I ask myself, "self", "What are you trying to do?" My answers were: e-mail, cloud storage,

private contacts and calendar, read the news, monitor the house when I'm out, listen to music, watch a movie or TV, and save my notes where I can find them.

Then I search the Internet for Open Source software that seems to do that. Sites like OpenSource [1], LinuxAlt [2], LinuxJournal [3], and GNU [4] below, are good starters.

Avoid sites with lots of advertising, binary downloads and non-secure web servers. Do not download packages unless the Linux package manager cannot find it. Just look to see if the things you want are available, what the name is, then do
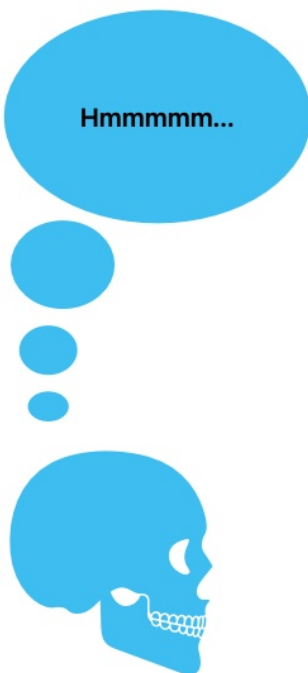
```
$ apt search <name>
```

to see if it is already packaged for you. This way you get secure software and it will be updated automatically.

1. https://opensource.com/alternatives
2. https://www.linuxalt.com/
3. https://www.linuxjournal.com/
4. https://www.gnu.org/software/software.html

Questions

Does it do what you want
Does it have an active community
When was the last release
Do they care about security
What language does it use
How does it fit on your existing hardware

Linux package distributions like Ubuntu [1], Debian [2], Fedora [3], and OpenSuse [4] are under active development and have support website links back to each project. Remember once you pick a Linux distribution, you need to stick with it's packages.

Smaller products can be found on github.com [5] and gitlab.com [6] and are generally compatible with any distribution.

Be Curious, Just Try

Take some time to read the use-case and

security pages. Passwords should be encrypted, network access should be none or limited, test cases should be extensive and developer participant counts made available. If you open this application to the internet, expect hackers to come at it, so error logging and monitoring is a must.

Find out what programming language(s) they use, because you should have a quick look at it to decide if it is readable and something you might tackle if needed. Otherwise be prepared to discard it if a big problem arises.

When you get some

1. https://packages.ubuntu.com
2. https://www.debian.org/distrib/packages
3. https://fedora.pkgs.org/
4. https://software.opensuse.org/explore
5. https://github.com/search
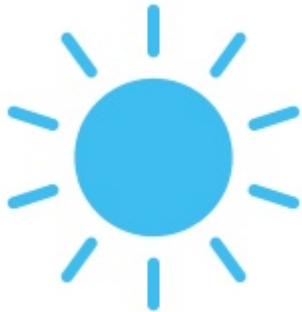6. https://gitlab.com/explore/projects

time, try learning bash [1], Python [2], C [3], C++ [4], and Rust [5]. That will have you covered for most projects, and open up new doors of fun.

Lastly think about what computer will run it, does it have enough disk, memory, cpu and backup methodologies.

Everyone with some computer skills should take control of your data and programs in your house, with a few minutes work every day to gain knowledge, security, and **improve your life**.

1. https://linuxconfig.org/bash-scripting-tutorial-for-beginners
2. https://www.learnpython.org/
3. https://www.cprogramming.com/
4. https://www.codecademy.com/learn/learn-c-plus-plus
5. https://google.github.io/comprehensive-rust/welcome.html

Tips

The man command is an interface to the system reference manuals

Find help with related man pages:
```
$ apropos -e mail
mail (1)            -process mail messages
mail.mailutils (1)-process mail messages
mailx (1)           -process mail messages
mbox (5)            -Format mail storage.
mutt (1)            -Mutt Mail UserAgent
muttrc (5)          -Configuration file for
the Mutt Mail User Agent
```

List manual pages that match a regular expression either in title or body:
```
$ man -k '\<copy\>.*archive'
cpio (1)            -copy files to and
from archives
```

Read the muttrc man page, using the man section number (optional if unique):
```
$ man 5 muttrc
<press q to quit>
```

Man Section Numbers

1 - Executable programs or shell commands
2 - System calls (functions provided by the kernel)
3 - Library calls (functions within program libraries)
4 - Special files (usually found in /dev)
5 - File formats and conventions, e.g. /etc/passwd
6 - Games
7 - Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8 - System administration commands (usually only for root)
9 - Kernel routines [Non standard]

```
Filesystem Location -> /usr/share/man/
```
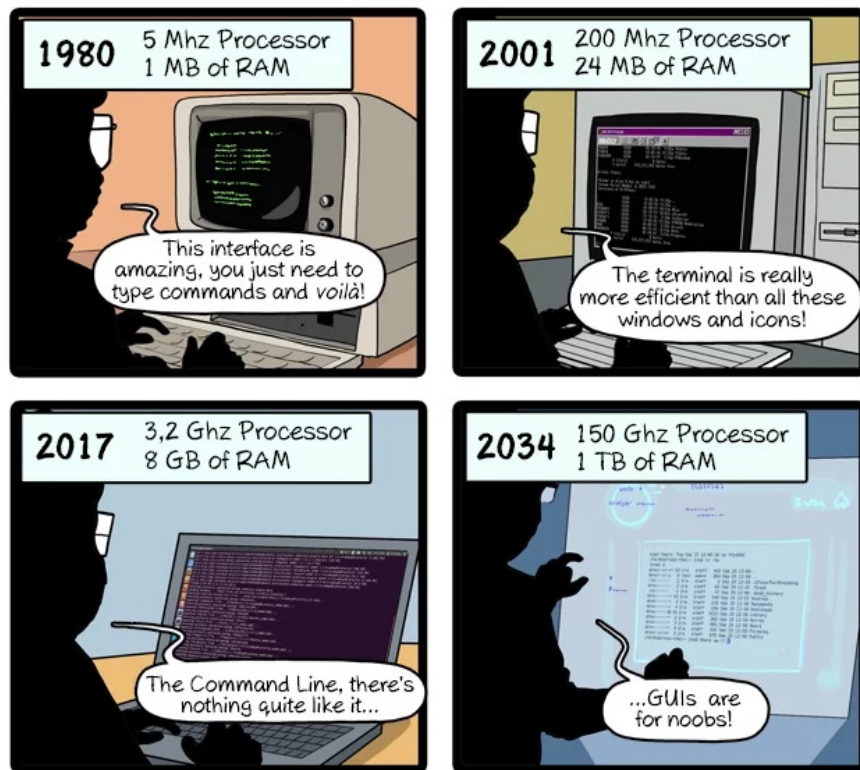
CommitStrip.com

Coming up in the next issue:

Setup a new Server

I use this as a guide to enable proper monitoring and maintenance of any new server on the network.

Created by: Don Cohoon

January 2025 | Issue: 1.0

https://dfcsoftware.github.io/linux-in-house/category/zines.html